

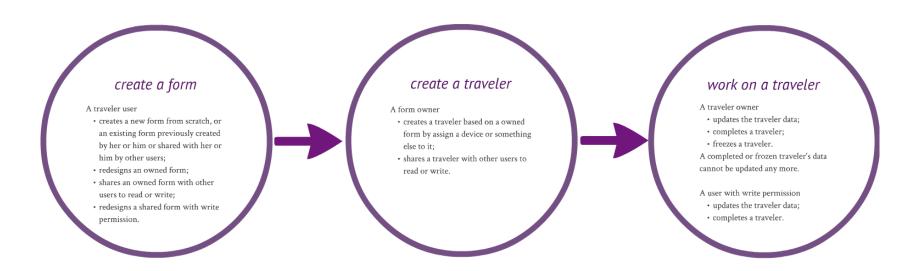
Traveler @ FRIB

Dong Liu (liud@frib.msu.edu) Software Engineer





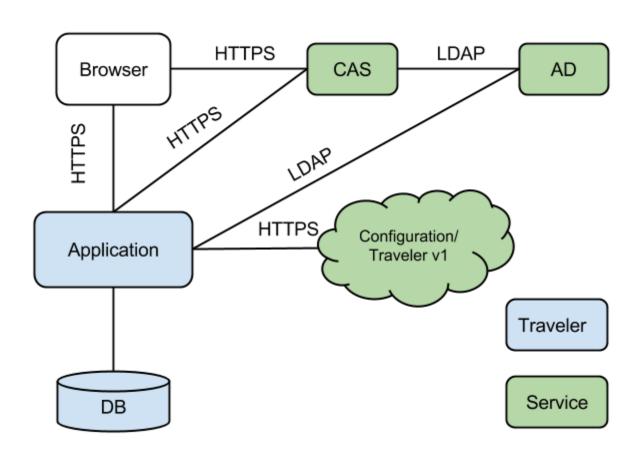
Traveler workflow



http://prezi.com/x_dsx1fpzboy/?utm_campaign=share&utm_medium=copy&rc=ex0share



Application architecture



Technology stack

- Node.js (https://nodejs.org/)
- Express.js (<u>http://expressjs.com/</u> 3.1.0 needs upgrade)
- bootstrap (<u>http://getbootstrap.com/</u> 2.3 needs upgrade)
 - Jade (http://jade-lang.com/) for template
- jQuery (<u>https://jquery.com/</u> 1.9)
- Datatables (https://www.datatables.net/ 1.9.4 needs upgrade)
- Twitter typeahead (http://twitter.github.io/typeahead.js/)
- Form binder (<u>https://github.com/dongliu/formbinder</u>)
- Form builder
 - Rivets.js (http://www.rivetsjs.com/)
- CAS for authentication (https://github.com/rubycas/rubycas-server)
- MongoDB/mongoose.js (http://mongoosejs.com/)

node + mongo

- JSON from DB to application to client
 - No ORM (++)
 - Hard to support other format (-)
- No threading and synchronized/.concurrent.
 - Almost no chance for racing/deadlock (+++)
 - Callback hell (--)
- Flexible data structure
 - Support array of anything (++)
 - Easy schema change and back compatible (+++)
 - Loose validation(--+)
 - Duplication and inconsistence (--+)

Paradigm shifts

- desktop app -> network and mobile app (client/server)
- local library -> network API
- local call -> RPC -> messaging, event-driven, REST
- synchronous -> asynchronous
- purely object-oriented -> functional
- basic client -> rich client
- heavy static framework -> light dynamic framework/language
- relational -> unstructured
- one-size-fits-all -> diversity

Design and development process

Resources

- Interface first for testing
- View -> Model -> Controller
 - » HTML/CSS -> test -> DOM/JavaScript -> test -> XHR -> test -> GET /model -> test -> POST/PUT /model -> test -> DELETE /model -> test
- Most resource loaded via JSON/HTTP -> API
- Responsive design
 - Better user experience on different screens
 - Mobile client friendly
- Reactive design
 - Render only partial page when possible
 - Lightweight model view viewmodel (MVVM)
 - Push lightweight processing tasks to the client

TODO



Manage and collaborate.

With forms, travelers, and reports. Need help?



- Restructure the application for reporting feature
 - User defined traveler groups for tracking and reporting
 - User interfaces for form/traveler/report

Traveler resources

Source

https://github.com/dongliu/traveler

Design documents

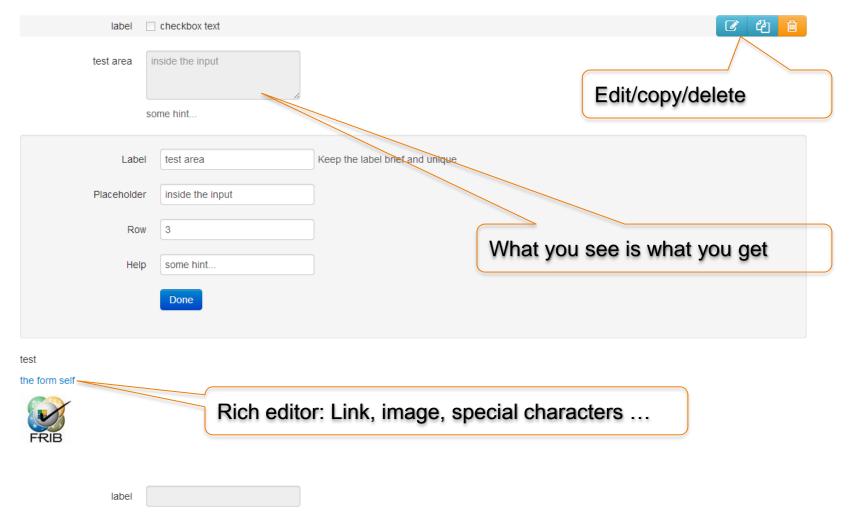
- https://drive.google.com/file/d/0B39LbIQJq1dreEJSaVJ2TGRCMzA
- https://drive.google.com/file/d/0B39LbIQJq1drRkE4YV9mVl84Wjg
- https://docs.google.com/document/d/12p4eO0b-Bv3jGVLsbYzMU0leYaxHX6r9DYDpn2Udrs/edit?usp=sharing

Manuals

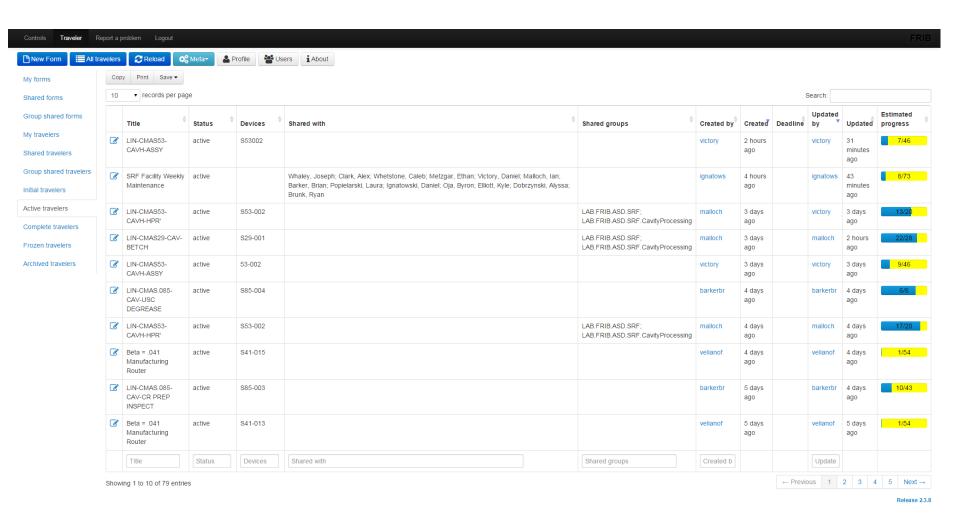
- https://docs.google.com/document/d/1I9YY2_czQ193zDv8mIhRMFYIpaMgf GyHr23EwrJ2FLM/pub
- http://rawgit.com/dongliu/traveler/master/public/about.html
- http://rawgit.com/dongliu/traveler/master/public/api.html

Form builder

test1

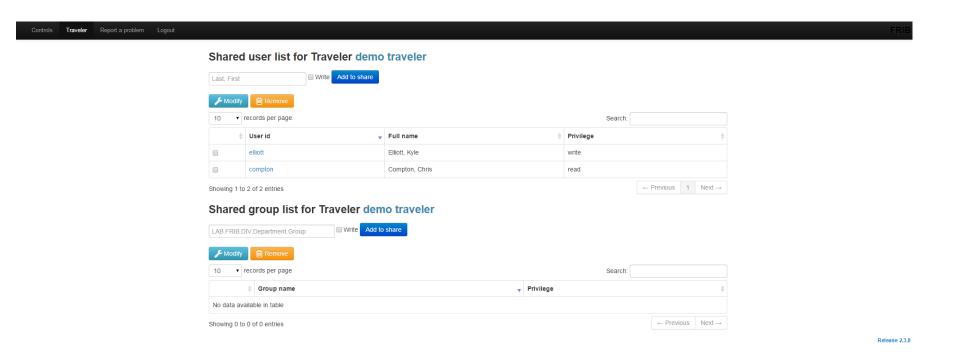


Main user interface

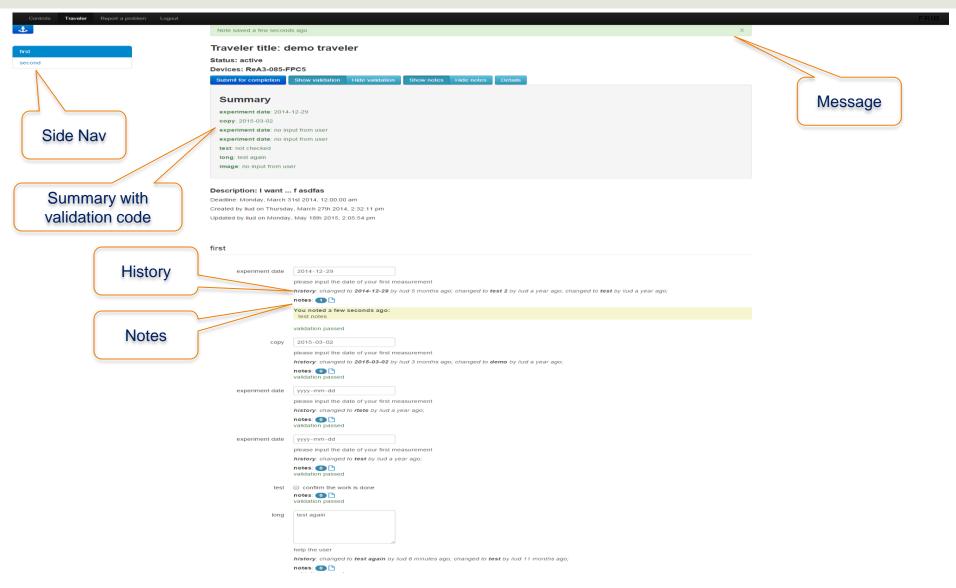


D. Liu -- EPICS Collaboration, May 2015, Slide 11

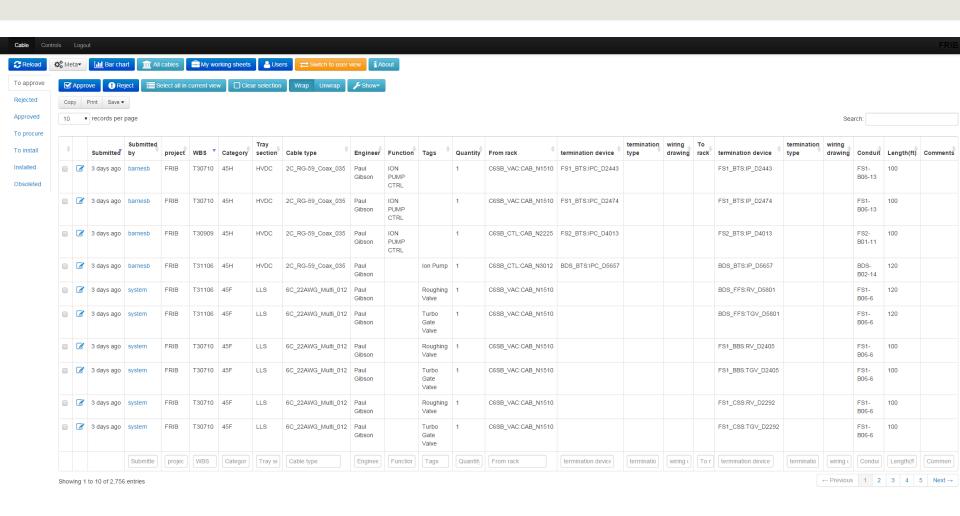
Traveler share



Traveler page



Cable main interface



Plot



User management

